


1. Fototechnik und digitale Bildbearbeitung

- 1.1 Grundlagen der Fototechnik
- 1.2 Digitale Fotografie
- 1.3 Einführung in die fotografische Bildgestaltung
- 1.4 Speicherung digitaler Bilddaten 
- 1.5 Bearbeitung digitaler Bilder
- 1.6 Programmierung von Bildbearbeitung mit Java

Literatur:

J.+R. Scheibel, Fotos digital – Basiswissen, vfv 2000

Auflösung (Pixelzahl)

Typische Zahlen im Jahr 2011:

Marktgängige Digitalkameras zwischen 10 Mio. und 24 Mio. Pixel

Preisgünstige "Consumer"-Kameras bei 12 Mio. Pixel

Semiprofessionelle Kameras (z.B. Canon EOS 50D) ca. 15 Mio. Pixel

Digitale Mittelformatkameras (z.B. Pentax 645D, Hasselblad H4D) mit 40-60 Mio. Pixel

Benötigte Auflösung (Pixelzahl) abhängig von geplanter Verwendung:

Richtwerte für Auflösung r in points per inch (ppi):

Monitor 72 ppi, Tintenstrahldrucker 200 dpi, Offsetdruck 225 dpi,

Belichtung auf Fotopapier 300 dpi

Berechnung der benötigten Auflösung pro cm^2 :

$$\text{Pixelzahl} = (r / 2,54)^2$$

Beispiel: Fotopapier (300 dpi) benötigt ca. 14.000 Pixel/ cm^2

Beispiele:

9 x 13 cm Papierbild benötigt ca. 1,6 Mio. Pixel

18 x 24 cm Papierbild benötigt ca. 6 Mio. Pixel

30 x 40 cm Papierbild benötigt ca. 16 Mio. Pixel

30 x 40 cm Monitorausgabe benötigt ca. 1 Mio. Pixel

Speicherbedarf für Bilder

Typische Bildtiefe: 1 Byte pro Farbe, d.h. 24 Bit/Pixel

6 Mio. Pixel Bildgrösse: 18 MByte Rohdaten pro Bild

Bildformate:

TIFF: verlustfrei, portabel, gross

Proprietäres Format der Kamera ("RAW" bei Canon, "NEF" bei Nikon)
mit verlustfreier (schwacher) Kompression (ca. 6:1)

Besonders geeignet für Nachbearbeitung (z.B. bezüglich Weißabgleich)

JPEG: verlustbehaftet, portabel, klein

Digitalkameras bieten meist

Wahlmöglichkeiten an:

Bildgröße

Auflösung

Wahl je nach Verwendungszweck

Beispiel:
Original 2,1 Mio. Pixel
Dateigrösse JPEG 640 kB



Klassifikation von Speichermedien

Speicherdauer:

permanent vs. temporär

Zugriffsart:

sequentiell vs. wahlfrei/adressierbar

Benutzungsmodus:

nur lesen vs. lesen und schreiben

Zuverlässigkeit:

Anzahl der zulässigen Schreib-/Lesezyklen

Generelle Ausfallwahrscheinlichkeit

Halbleiterspeicher: Übersicht (1)

DRAM: Dynamic Random Access Memory

Prinzip: Gezielt ansprechbare grosse Menge von Kleinstkondensatoren

Kondensator speichert elektrische Ladung

Benötigt regelmässige Auffrischung (*refresh*) des Speicherinhalts

Basis für Arbeitsspeicher aller modernen Computer

SRAM: Static Random Access Memory

Prinzip: Information als Zustand von bistabilen Schaltungen (Flip-Flops)

Inhalt bleibt erhalten solange Versorgungsspannung anliegt

Relativ geringe Kapazität, schneller Zugriff

Verwendung z.B. in Zwischenpuffern (*Cache*)

ROM: Read-Only Memory

Prinzip: Information durch feste "Verdrahtung" dargestellt

PROM: Programmable ROM

Prinzip: Dauerhafte (einmalige) Programmierung, etwa durch "Durchbrennen" von Leitungen

Halbleiterspeicher: Übersicht (2)

EPROM: Erasable Programmable ROM

Prinzip: Laden eines sogenannten "Floating Gate"
in einem Feldeffekt-Transistor
(Ladung erhält sich mindestens 10 Jahre)

Löschen des Speichers durch UV-Licht möglich



EPROM

EEPROM: Electrically Erasable PROM

Prinzip: Floating Gate durch lange und starke Spannungsimpulse aufladbar
und (bei umgekehrter Polarität) entladbar

Typischerweise geringe Kapazität (mehrere KByte) und lange Schreibzeiten

Flash-Memory

Analog zu EEPROM, dünneres Tunneloxid, geringere Programmierspannung,
blockweises Löschen, ca. 10.000 Programmierzyklen möglich

Sehr robuster und permanenter (ca. 10 Jahre) Wechselspeicher

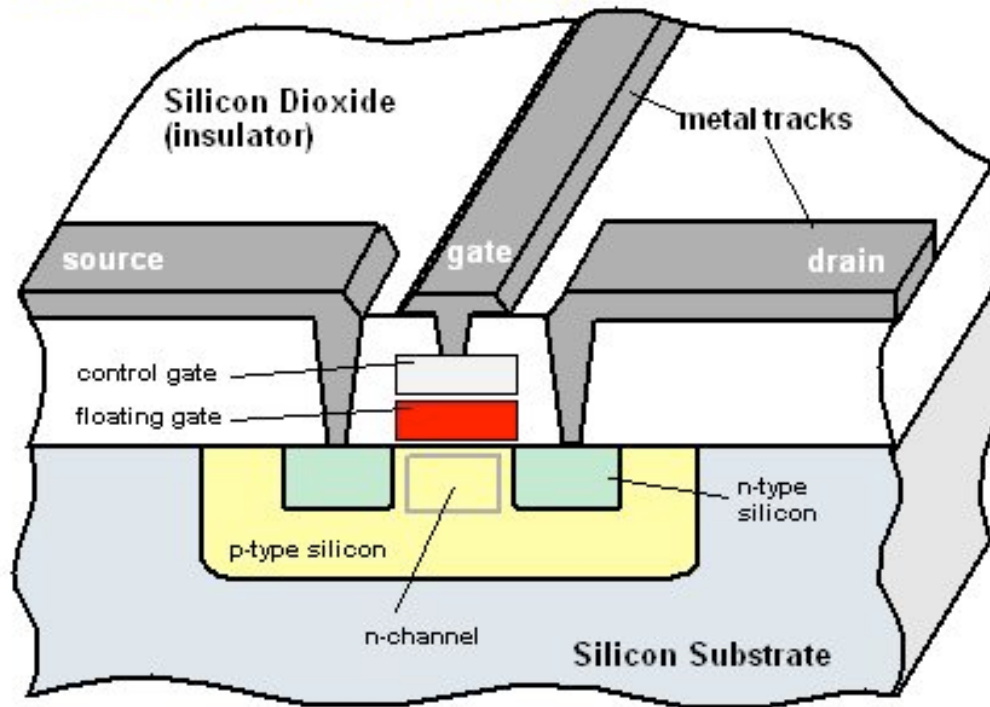
Praktische Erscheinungsformen: Flash Memory Devices, Flash-Speicherkarten

Früher langsam (400-800 KByte/s),
heute deutlich schneller: bis zu 100 MByte/s lesen + schreiben

Floating Gate Transistor (in EEPROM/Flash)

From Computer Desktop Encyclopedia
© 2005 The Computer Language Co., Inc.

EEPROM and Flash Transistor



Typen von Flash-Speicherkarten

Compact Flash (CF) - derzeit bis zu 128 GB

Typ I: dünner (43 x 36 x 3,3 mm)

Typ II: dicker (5 mm), auch Mikro-Festplatte ("MicroDrive")

Controller in Karte integriert (Kompatibilität mit alten Kameras)

SmartMedia (SSFDC):

superdünn (0,8 mm), bis 128 MB

Frühes, jetzt aber auslaufendes Format

xD Picture Card (Olympus, Fuji)

Als Nachfolger von SmartMedia geplant, bis 8 GB

SD Card/ MM Card (Secure Digital Card):

klein (32 x 24 x 2,1 mm), ursprünglich bis zu 2 GB

Controller in Karte integriert

SDHC = High Capacity Version, bis zu 32 GB

SDXC = Extended Capacity Version, bis zu 2 TB

Speed-Klassen (ab SDHC): Zahl ist Übertragungsrate in MB/s

Versionen mit Rechtemanagement (meist unwichtig für Fotografie)

MemoryStick/Memory Stick Pro (Sony):

klein (50 x 21,5 x 2,8 mm), proprietär

Halbformat: "MemoryStick Duo"

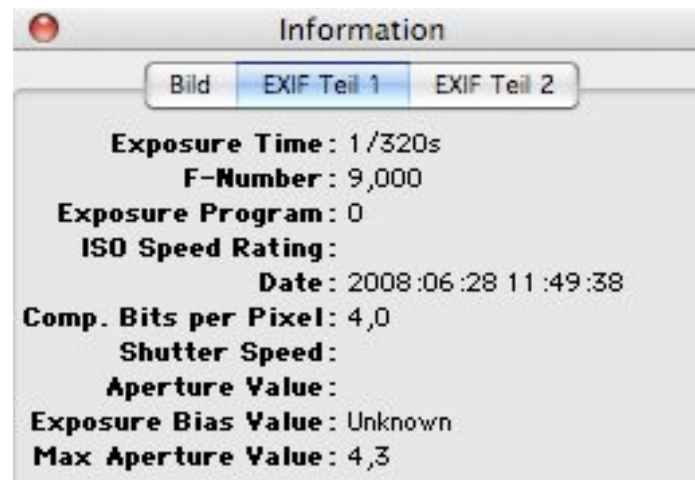


EXIF

EXIF = Exchangeable Image File Format

Ablage der meisten zum Aufnahmezeitpunkt vorliegenden Daten in der Bilddatei

Achtung: Daten verschwinden meist bei weiterer Konversion/Bearbeitung des Bilds!



Manchmal APEX-Werte (additiv):

Blendenwerte: APEX 0 = f, APEX 1 = f/1,4, APEX 4 = f/4, ...

Zeitwerte: APEX 0 = 1s, APEX 1 = 1/2s, APEX 9 = 1/500s, ...

Aufgenommen:	28.06.2008 11:49:38
Digitalisiert:	28.06.2008 11:49:38
▼ Datei	
Name:	DSC_0075.JPG
Größe:	2,5 MB
Geändert:	28.06.2008 12:49:38
Importiert:	28.06.2008 22:55:05
▼ Kamera	
Hersteller:	NIKON CORPORATION
Modell:	NIKON D70
Software:	Ver.1.03
▼ Belichtung	
Verschlusszeit:	1/320
Blende:	f/9,0
Max. Blende:	f/4,4
Belichtungskorrektur:	0,00
Belichtung:	Nicht definiert
Belichtungsindex:	—
Brennweite:	46 mm
Entfernung:	—
Abtastung:	Ein Farb-CCD
Lichtquelle:	Unbekannt
Blitz:	Aus
Messung:	Mehrfeld
Helligkeit:	—
ISO-Einstellung:	800
GPS-Breitengrad:	—
GPS-Längengrad:	—
GPS-Höhe:	—

DCF-Dateistruktur

DCF = Design Rule for Camera File System

- JEITA Standard
- Dateinamen und Ordernamen auf den Speichermedien für Kameras

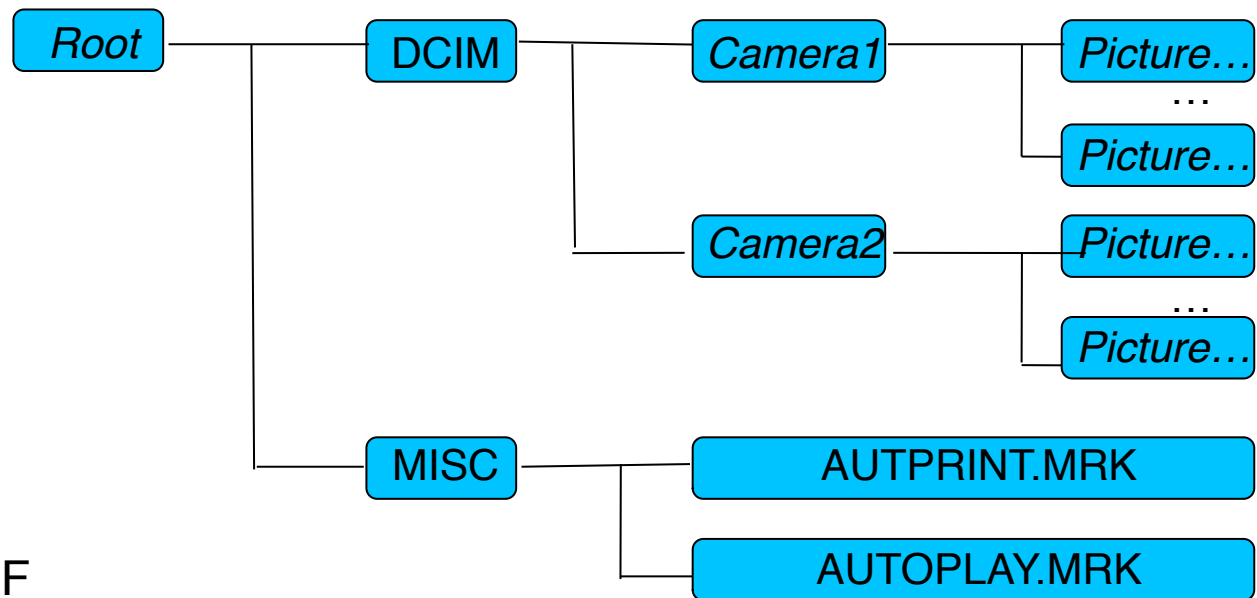
Kameraordner: NNNAAAAA

- *NNN* Ziffern (verschieden innerhalb DCIM), Rest frei

Bilddateien: AAAANNNN

- *AAAA* Großbuchstaben (frei), *NNNN* Laufende Nummer

Ordnerstruktur:



<http://www.exif.org/dcf.PDF>

DPOF

Datenformat zur Ergänzung der Bilddaten um Auftragsdaten für
Bearbeitungsaufträge


DPOF = Digital Print Order Format

Vorwiegend für den Endverbraucher-Markt

Ermöglicht es, an der Kamera bzw. in einem Bildbearbeitungsprogramm
festzulegen:

- Bildnummer, Option für Index-Print ("Kontaktabzug"),
Anzahl der gewünschten Abzüge/Prints,
Formate der Abzüge, Adressdaten
- Integriert mit DCF-Dateistruktur (MISC-Ordner)

1. Fototechnik und digitale Bildbearbeitung

- 1.1 Grundlagen der Fototechnik
- 1.2 Digitale Fotografie
- 1.3 Einführung in die fotografische Bildgestaltung
- 1.4 Speicherung digitaler Bilddaten
- 1.5 Bearbeitung digitaler Bilder 
- 1.6 Programmierung von Bildbearbeitung mit Java

Literatur:

B. + U. Steinmüller, Die digitale Dunkelkammer, dpunkt 2004
<http://www.outbackphoto.com>

Die klassische Dunkelkammer

“Dunkelkammer” (*darkroom*):

Dunkel bis auf Speziallicht (meist rot)

Vom belichteten Film zum Papierabzug:

Entwickeln des Films: liefert Negativ (Original)

“Vergrößern”:

Projizieren auf lichtempfindliches Papier

Belichten mit genau definierter Belichtungszeit

Entwickeln, fixieren, trocknen des Papierabzugs

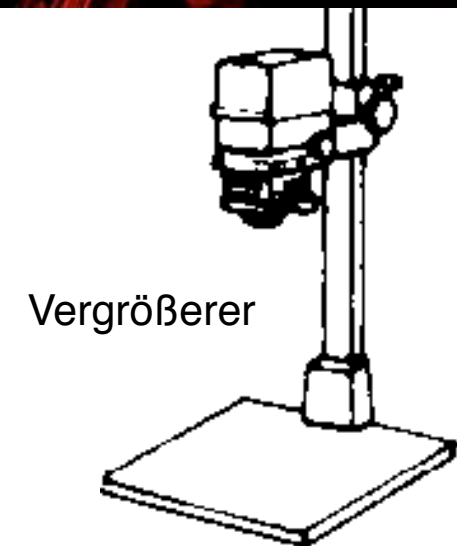
Vielfältige Gestaltungsmöglichkeiten:

Ausschnittwahl

Filter (Farbkorrekturen, Effekte)

Belichtung

Selektive Belichtung durch Abdecken von Partien (z.B. Aufhellen)



Die digitale Dunkelkammer

“Negativ” = Abtastergebnis des Kamerasensors

Bei (semi-)professioneller Arbeit meist im Rohformat (“RAW”)

Weiterbearbeitung mit Software

Ausschnittwahl

Filter (Farbkorrekturen, Effekte)

Aufhellen, abdunkeln

Schärfe verbessern, weichzeichnen

Fehler der Aufnahmegерäte korrigieren

Weißabgleich

...

Erstellen des Papierbilds

Ausdruck bzw. Belichten auf Spezialpapier

Zweck der digitalen Bildbearbeitung:

Optimierung des Bilds für subjektive Wirkung

Als Spezialfall: Schaffung neuer Bilder

Software-Werkzeuge

Bildbearbeitungs-Software

Adobe Photoshop (hier verwendete Version: 7.0)

The Gimp (Open Source)

Raw-Konverter (für professionelles Arbeiten)

Adobe Camera Raw (Photoshop Plugin)

Bemühung um standardisiertes Raw-Format (DNG Digital Negative)

Proprietäre Software von Kamera-Herstellern

Z.B. Nikon Capture, Canon Digital Photo Professional, ...

open source: dcrw

Integrierte Nachbearbeitungs-Arbeitsplätze (digitale Dunkelkammer)

Apple Aperture, Adobe Photoshop Lightroom, Capture One, Bibble, ...?

Bild-Browser, Bild-Datenbanken

Utilities

Z.B. zum systematischen Benennen von Bilddateien

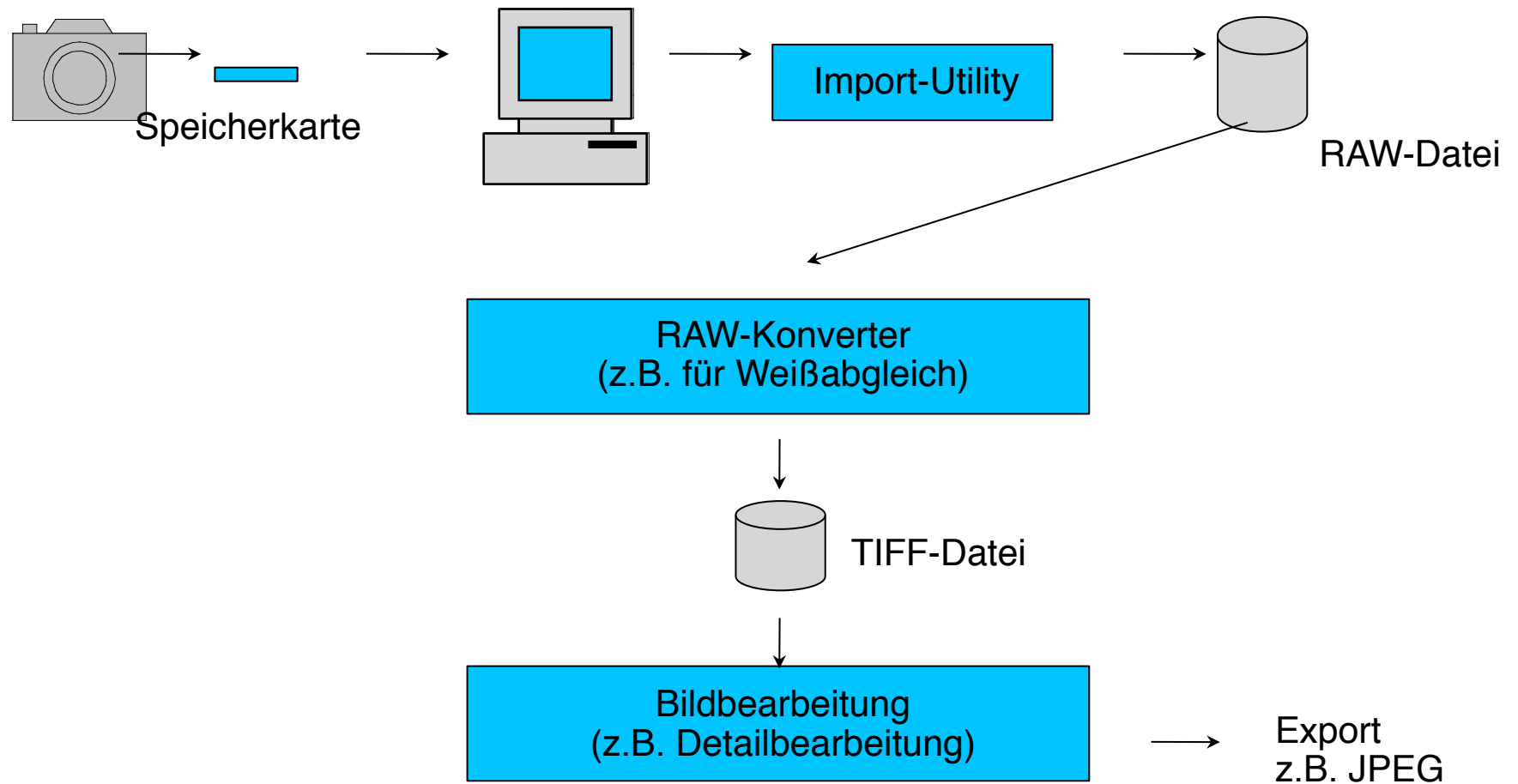
Integrierte Anwendungen

Z.B. Apple iPhoto

Grundlegender Arbeitsablauf der Bildbearbeitung



Konkretes Beispiel: Ablauf in der Praxis



JPEG oder RAW als Aufnahmeformat?

Vorteile von RAW:

- Kein Informationsverlust gegenüber der Aufnahmesituation
- Optimale Basis für Nachbearbeitung
- 16 Bit Farbtiefe unterstützt

Nachteile von RAW:

- Sehr groß
- Nur in teuren Kameras unterstützt
- Nachbearbeitung ist zeitaufwändig

Vorteile von JPEG:

- Kompakte Dateien
- Direkt für Papierbilder, Präsentation nutzbar
- Automatische Tonwertkorrektur, Weißabgleich etc.

Nachteile von JPEG:

- Verluste, Artefakte
- Meist nur 8 Bit Farbtiefe

Color-Management (CM)

Problem: Jedes Gerät hat unterschiedliche Randbedingungen der Farbdarstellung

Gerätespezifisches Farbspektrum: Farbraum oder Farb-*Gamut*

ICC (International Color Consortium): *Profil*-Beschreibungssprache

ICC-Profile für Geräte vom Hersteller verfügbar

Farbkalibrierung von Monitoren:

Einfache Softwarekalibrierung (Benutzerdialog)

Hardwarekalibrierung mit Messgerät

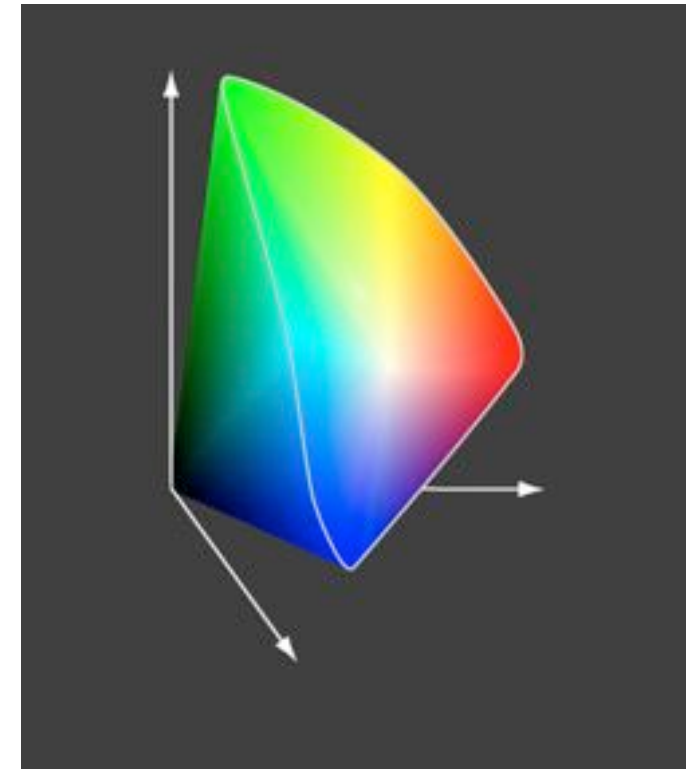
Liefert individuelle Geräteprofile

Austauschbarkeit von Farbdateien über
Standard-Profile

Definiert von ICC

Farbwerte immer relativ zum verwendeten Farbraum

Farbraum + Farbprofil liefert absoluten Farbwert
(z.B. für Monitor)



Verbreitete Standard-Farbräume

sRGB

Für die Bildschirmdarstellung ausgelegt
Kleiner als typischer Farbraum von
Digitalkameras

Adobe RGB (1998):

Verbreiteter Standard für Fotobearbeitung
Größer als sRGB, fast alle druckbaren
Farben

ProPhoto RGB

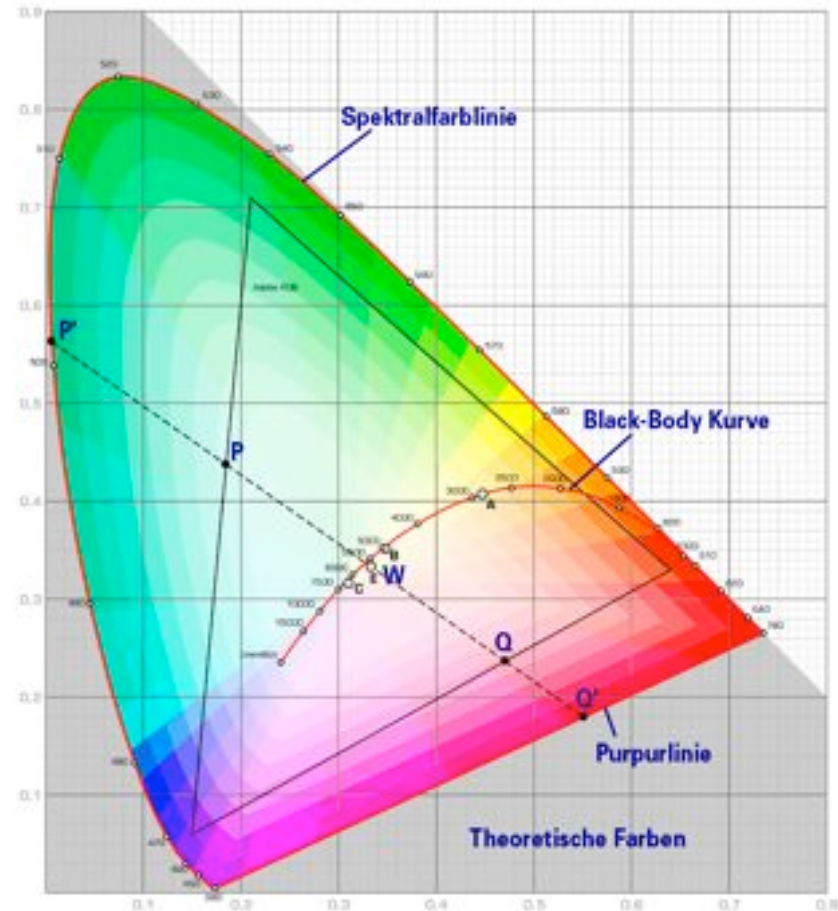
Kodak-spezifischer Standard mit großem
Farbumfang

Apple RGB

Apple-spezifisch, kleiner als Adobe RGB,
größer als sRGB

ECI-RGB

Speziell für Buch- und Zeitschriftendruck
(Druckvorstufe)



Globale Farbkorrekturen: Weißabgleich

Weißabgleich wird nötig durch verschiedene Spektral-zusammensetzungen, die subjektiv als „weiß“ empfunden werden

Ziel beim Weißabgleich ist die subjektive Wahrnehmung unter Berücksichtigung von Stimmungen

Weißabgleich ist möglich

In der Kamera

Im Raw-Konverter

In der Nachbearbeitung (nur eingeschränkt)

Methoden bei der Aufnahme:

Automatischer Abgleich

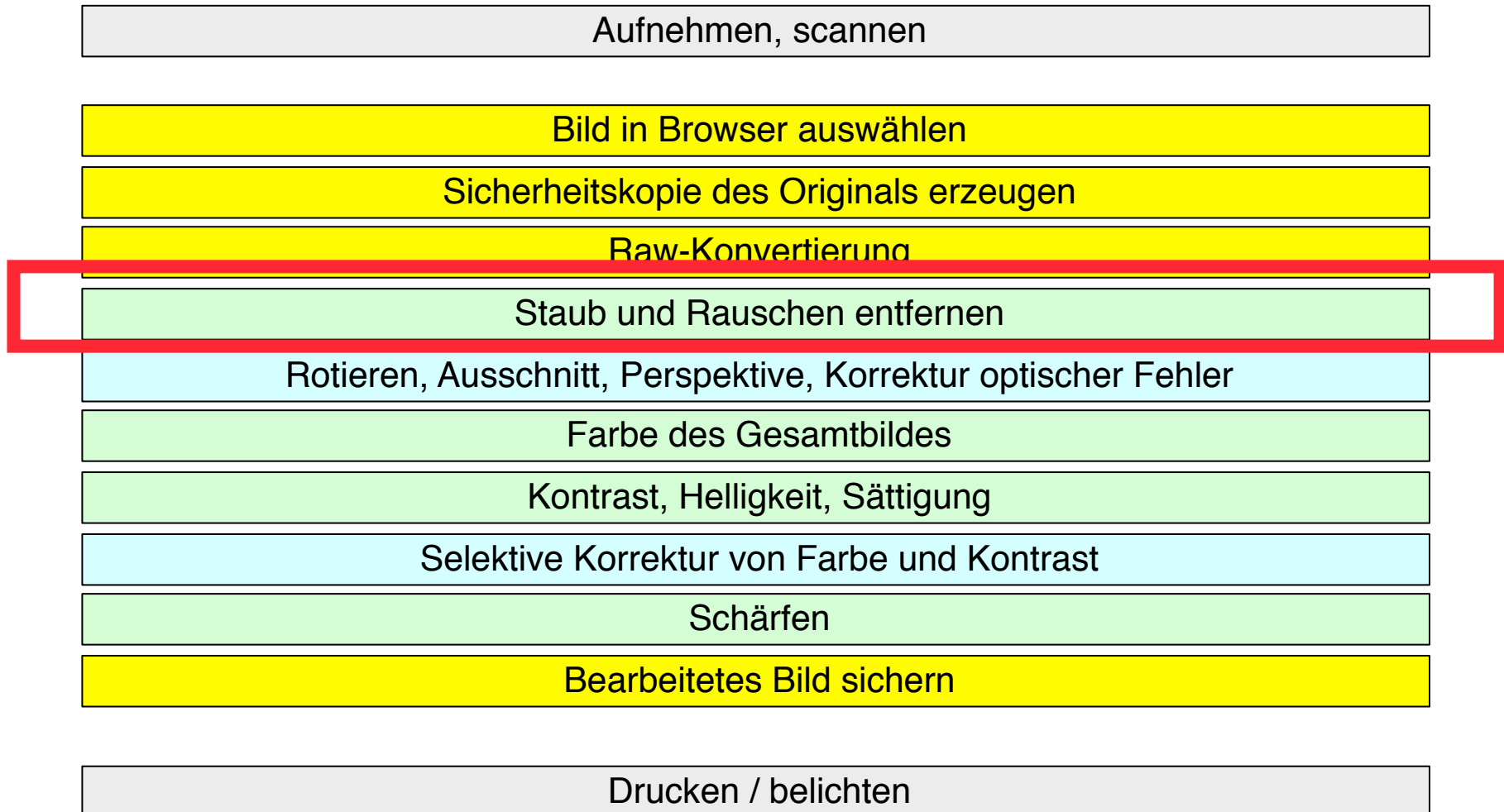
Verwendung eines grauen Bildelements als Referenz
(bei Aufnahme: Graue Standard-Farbkartons)

Methoden bei der Raw-Konversion:

Manuelle Einstellung



Grundlegender Arbeitsablauf der Bildbearbeitung



Störungsfilter



Rauschen oder Störungen

Staub, Kratzer, Digitalisierungsartefakte

In der Nachbearbeitung:

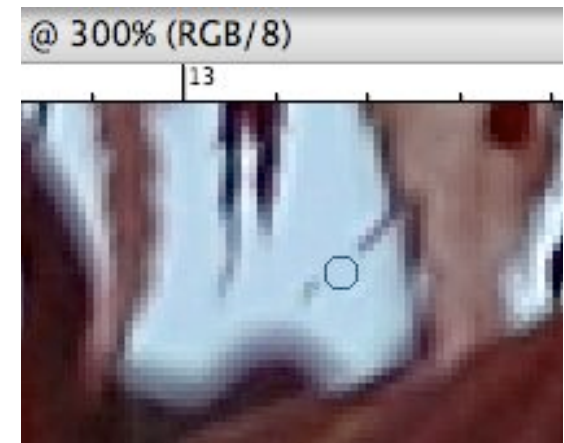
Manuell durch Retusche-Techniken entfernen
(z.B. Pinsel, Stempel)

Automatische Filter mit erheblichen Nebenwirkungen
(Weichzeichnung, Detailverlust):
z.B. Photoshop-Filter „Staub und Kratzer entfernen“

Im Raw-Konverter:

Raw-Konverter: Staubentfernung mit Referenzbild (z.B. Nikon Capture)

Optimal aber aufwändig



Grundlegender Arbeitsablauf der Bildbearbeitung



Geometrische Korrekturen

Kamera schief gehalten?

Horizont gerade (horizontal ;-) machen mit dem Messwerkzeug

Perspektivenkorrektur

Korrektur von „stürzenden Linien“

Wichtig in Kombination mit Weitwinkel-Objektiven

Bsp. Photoshop: (Auswahl), Bearbeiten->Transformieren->Perspektivisch verzerren

Korrektur von Linsenverzerrungen

Objektive haben konstruktionsbedingte und bekannte Verzeichnungsfehler

Spezialsoftware z.B. *PowerRetouche Lens Corrector*

Korrektur von Belichtungsfehlern durch Objektive

Z.B. „Vignettierung“

Oft bei herstellerspezifischen Raw-Konvertern möglich

Berücksichtigt Brennweite, Blenden- und Entfernungseinstellung zum Aufnahmezeitpunkt sowie Objektivtyp



Ausschnitt wählen



Bildaufbau kann hier nochmal grundlegend verändert werden!

Bildaufteilung

Störende Objekte

Was hier weggeschnitten wird, geht später auch bei Helligkeit+Kontrast nicht mehr ein!

Grundlegender Arbeitsablauf der Bildbearbeitung



Histogramm der Helligkeitsverteilung

Histogramm:

Allgemeines Konzept der mathematischen Statistik

Visualisierung der Häufigkeitsverteilung eines Merkmalwertes

Histogramme von Bildern:

x-Achse: Helligkeit oder Tonwert

Z.B. links dunkel, rechts hell

y-Achse: Anzahl der Pixel mit betreffendem Wert

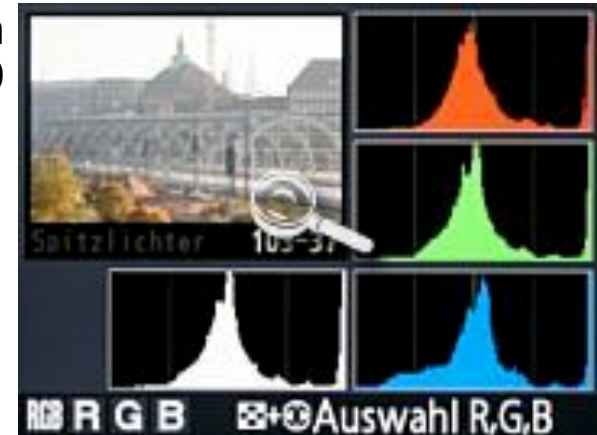
Anzeige:

Kamera

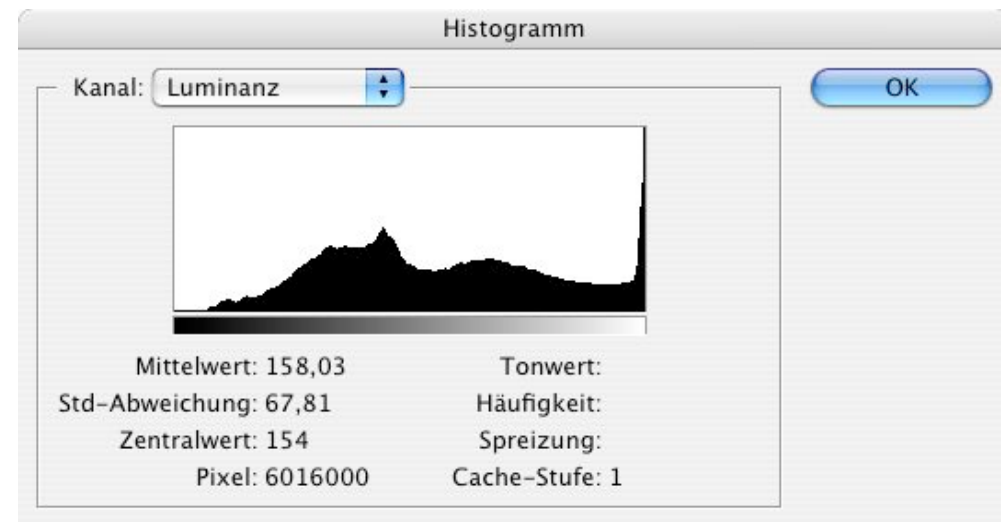
RAW-Konverter

Bildbearbeitungs-
Programm

Nikon
D200



Adobe Photoshop 7



Bildbeurteilung mit Histogrammen

Begriffe:

- *Tiefen* (= niedrige, dunkle Werte), *Lichter* (= hohe, helle Werte), *Mitteltöne*
- *Low-Key*, *High-Key*, *Average-Key*: Schwerpunkt der Werte

Ideale Verteilung:

Kompletter Wertebereich besetzt, klarer Anstieg/Abstieg zu den Rändern

Luminanz-Kanal:

Unterbelichtung (zu geringe Zeichnung in Tiefen) bzw. Überbelichtung (zu geringe Zeichnung in Höhen): „Gebirge“ am Rand abgeschnitten

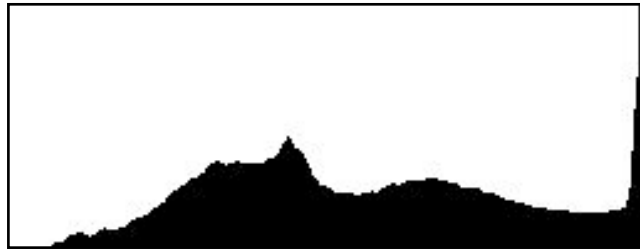
Geringer Kontrast: Schmalere Bereich der Werteskala ausgenutzt

Farbkanäle:

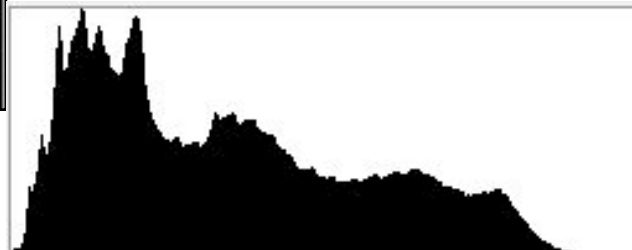
Gelegentlich Beschneidung nur eines Farbkanals (bei besonders „einfarbigem“ Motiven)

Im Luminanz-Histogramm nicht zu erkennen

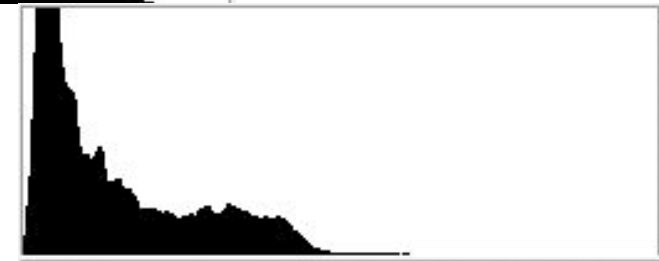
Beispiele zu Histogrammen



Überbelichtung



Normale Belichtung



Unterbelichtung



Tonwertumfang nicht ausgenutzt
(geringer Kontrast)

Abwedeln, Nachbelichten etc.

Selektive Helligkeitskorrektur:

Bildpartien zu hell: Nachbelichten

Bildpartien zu dunkel: „Abwedeln“

Namen von klassischer Labortechnik entlehnt

Pappmasken wie hier skizziert, müssen immer in Bewegung sein

Wirkungsvoll bei hoher Pixeltiefe des Originals

Abwedeln idealerweise bei Photoshop über eigene Ebene

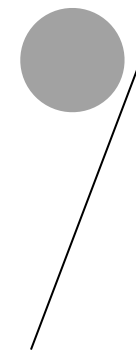
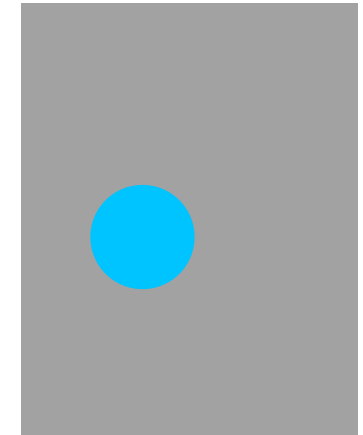
(sh. Steinmüller S. 135)

Weitere ähnliche Hilfsmittel:

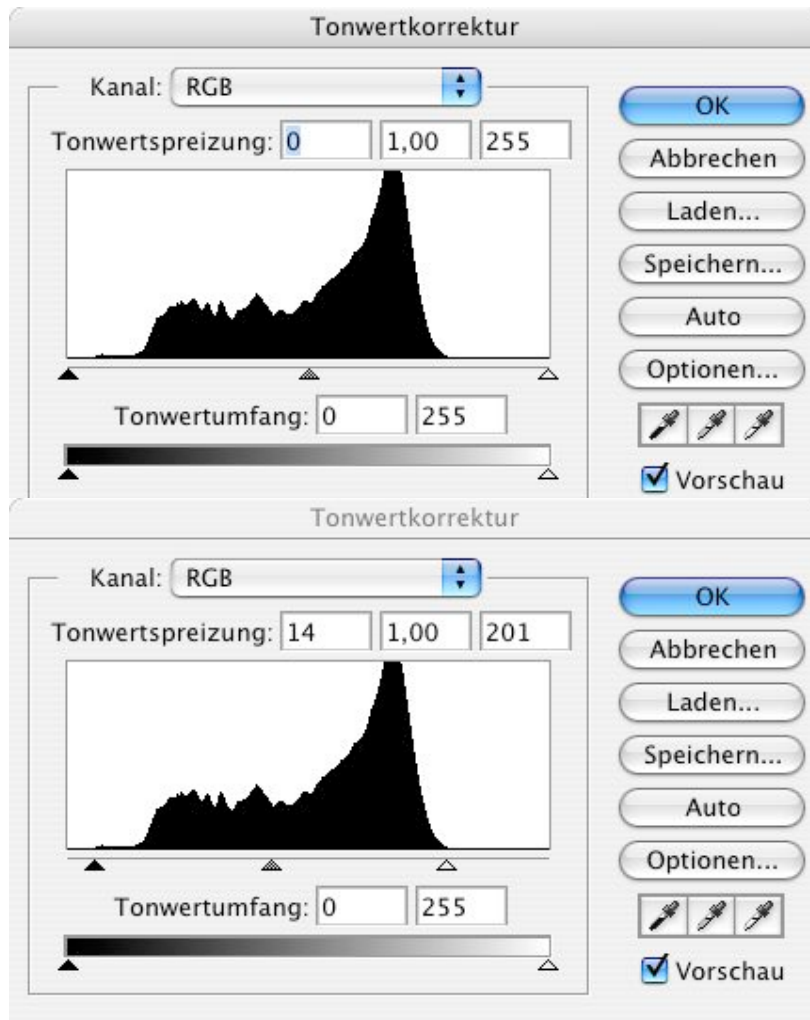
Selektive Weichzeichnung und Scharfzeichnung

Schwamm

Wischfinger



Tonwertumfang/Kontrast



Wertebereiche, die im Bild nicht vorkommen, können (oft) weggelassen werden

Spreizung des Intervalls vorhandener Werte auf Darstellungstiefe

- *Weißpunkt*: Grenze zu hellen Werten
- *Schwarzpunkt*: Grenze zu dunklen Werten

Tipp zu Photoshop: Drücken der „Alt“-Taste macht die weggeschnittenen Bildteile optisch sichtbar

Zu starken Kontrast vermeiden!

Gesamthelligkeit nachjustieren

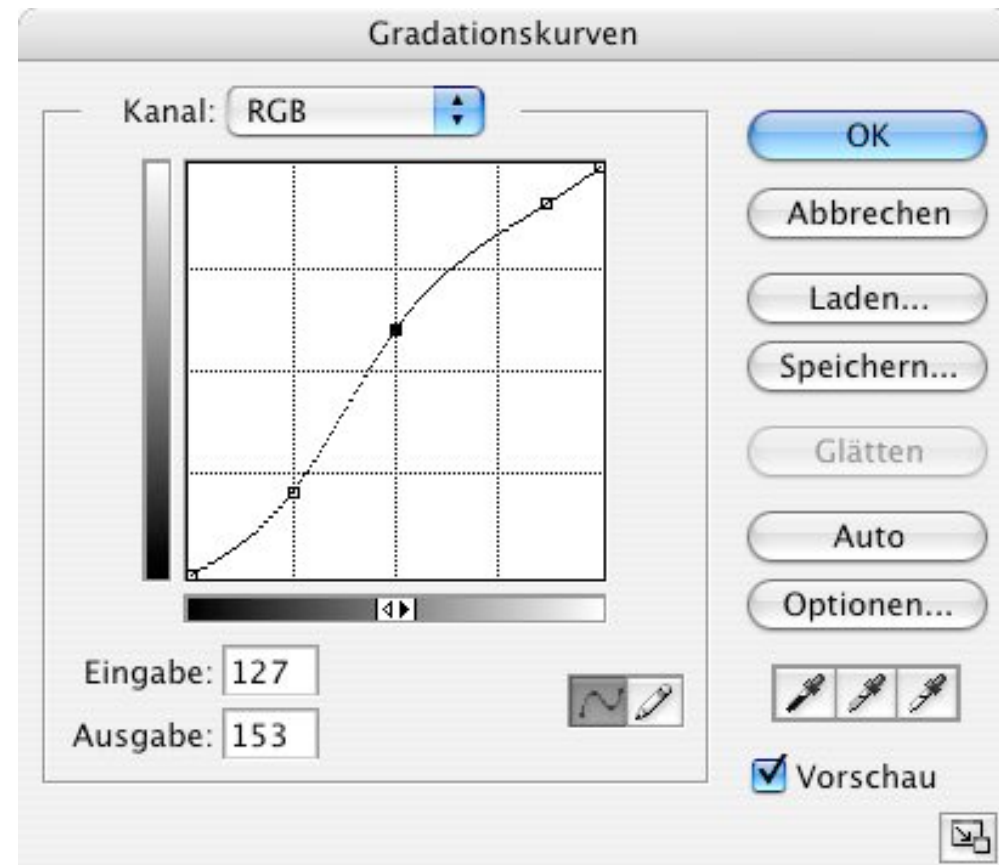
Gradationskurven

Statt einer linearen Spreizung ist oft eine nicht-lineare Modifikation der Tonwerte sinnvoll

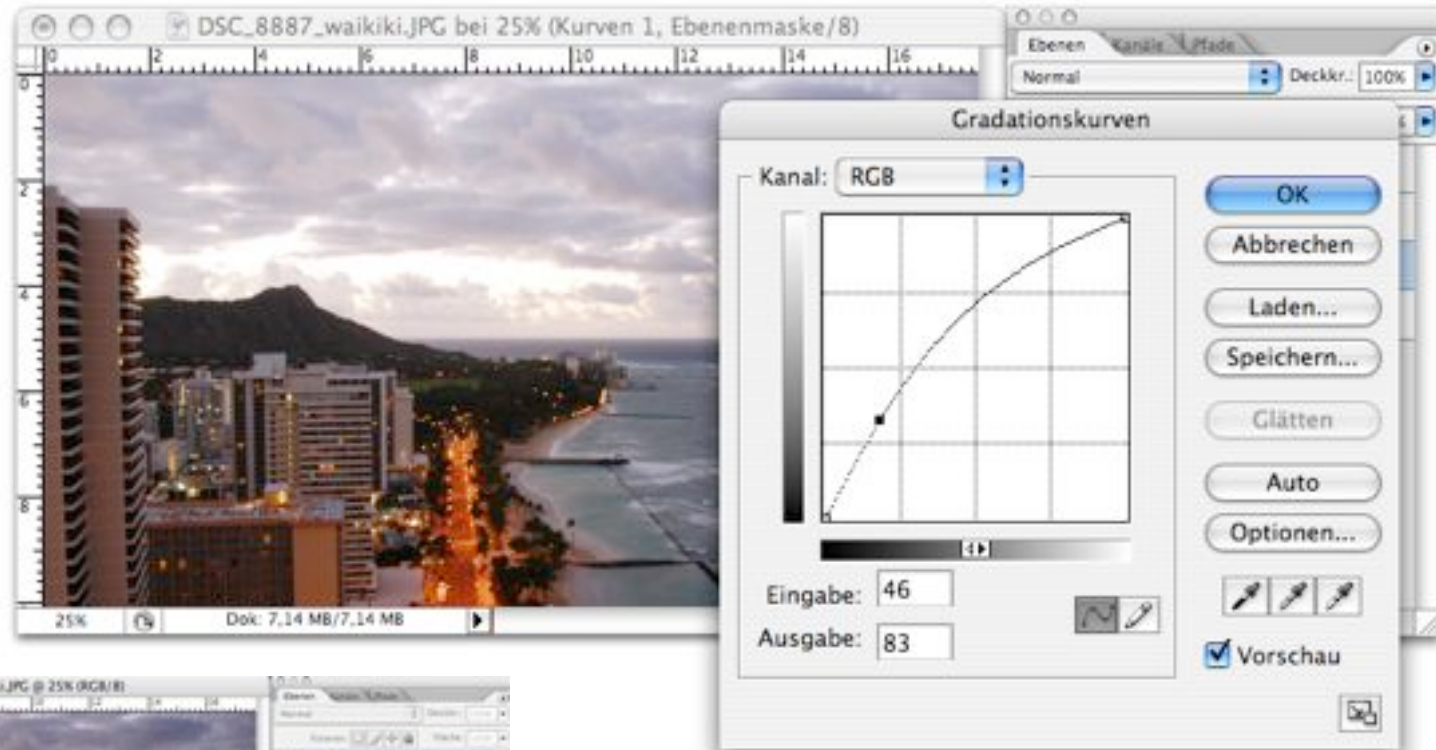
Gradationskurven erlauben die interaktive Modifikation mit direkter Vorschau

Oft sinnvoll: Leichte S-Kurve (entspricht dem Verhalten von fotografischem Film)

Photoshop: Auswahl von Hilfspunkten über Bilddarstellung (via Pipetten-Werkzeug) möglich

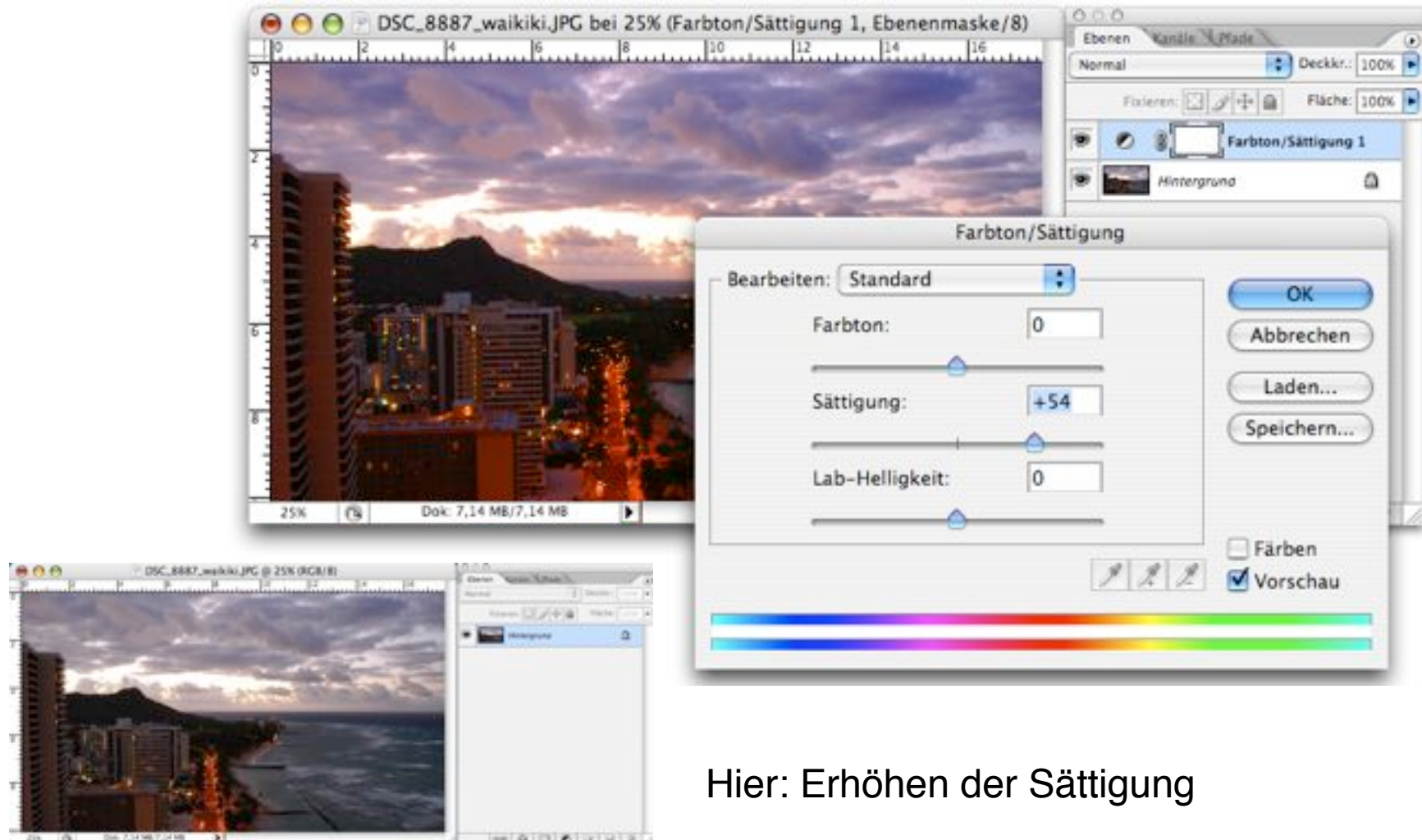


Gradationskurven



Hier: Aufhellen der dunklen Bildbereiche

Farbton + Sättigung



Hier: Erhöhen der Sättigung

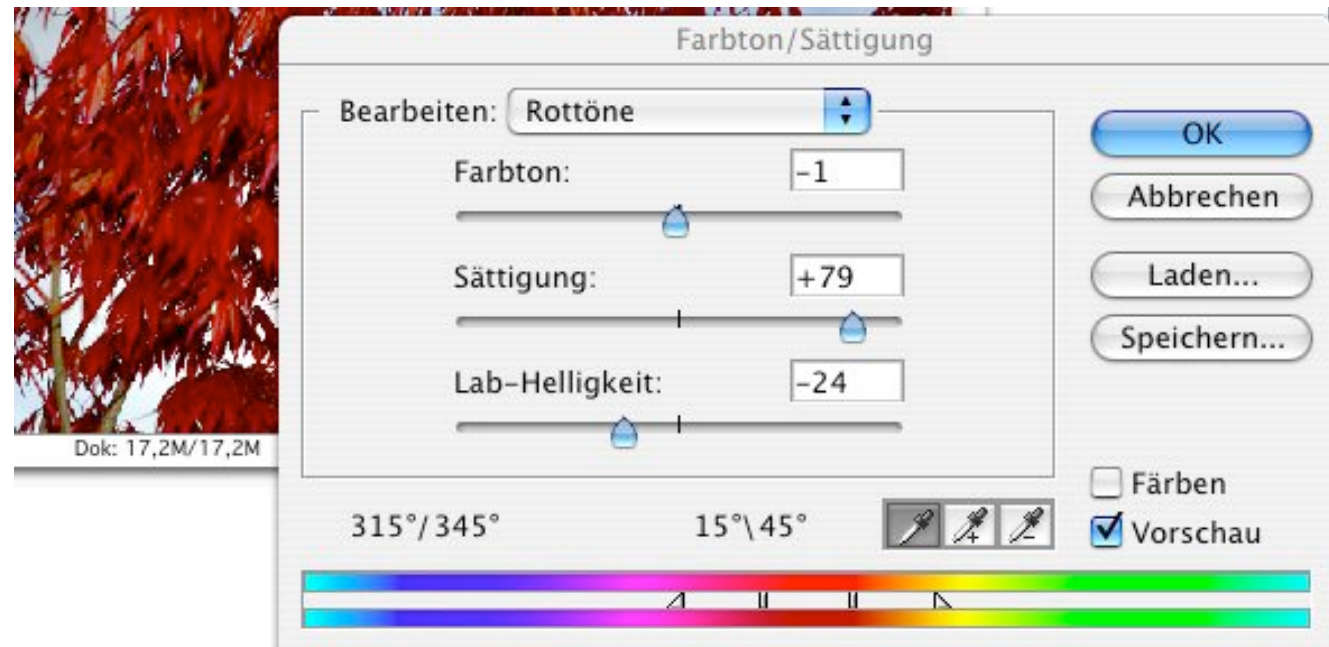
Selektive Farbkorrekturen

Z.B. mit „Farbton/Sättigung“ in Photoshop selektive Änderung einzelner Farbtöne möglich:

Himmel, Hauttöne, Grüntöne

Meist Erhöhung der Sättigung

Sollte unbedingt erst **nach** der korrekten Kontrasteinstellung erfolgen!

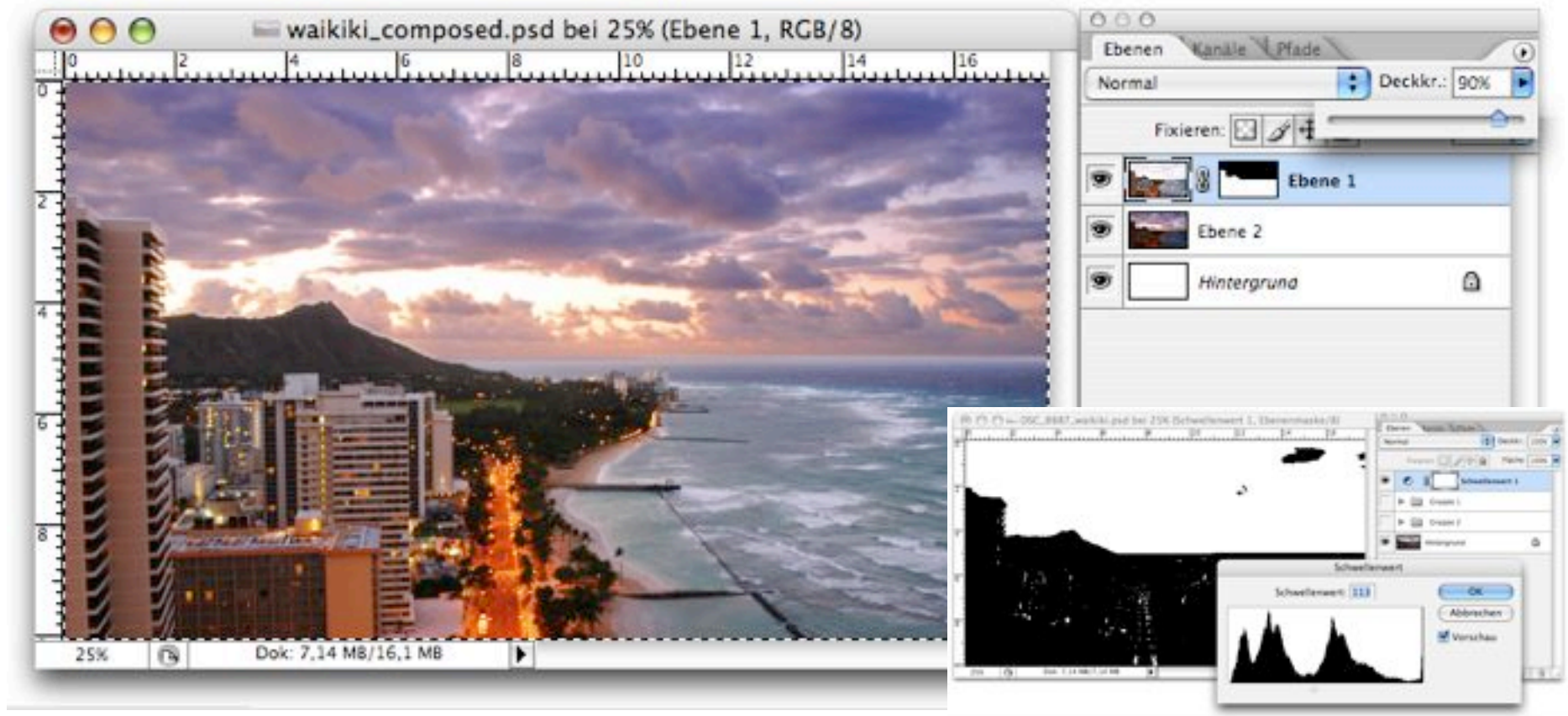


Mischen verschiedener Korrekturen



Problem hier: Trennlinie ist nicht gerade!

Ebenenmasken

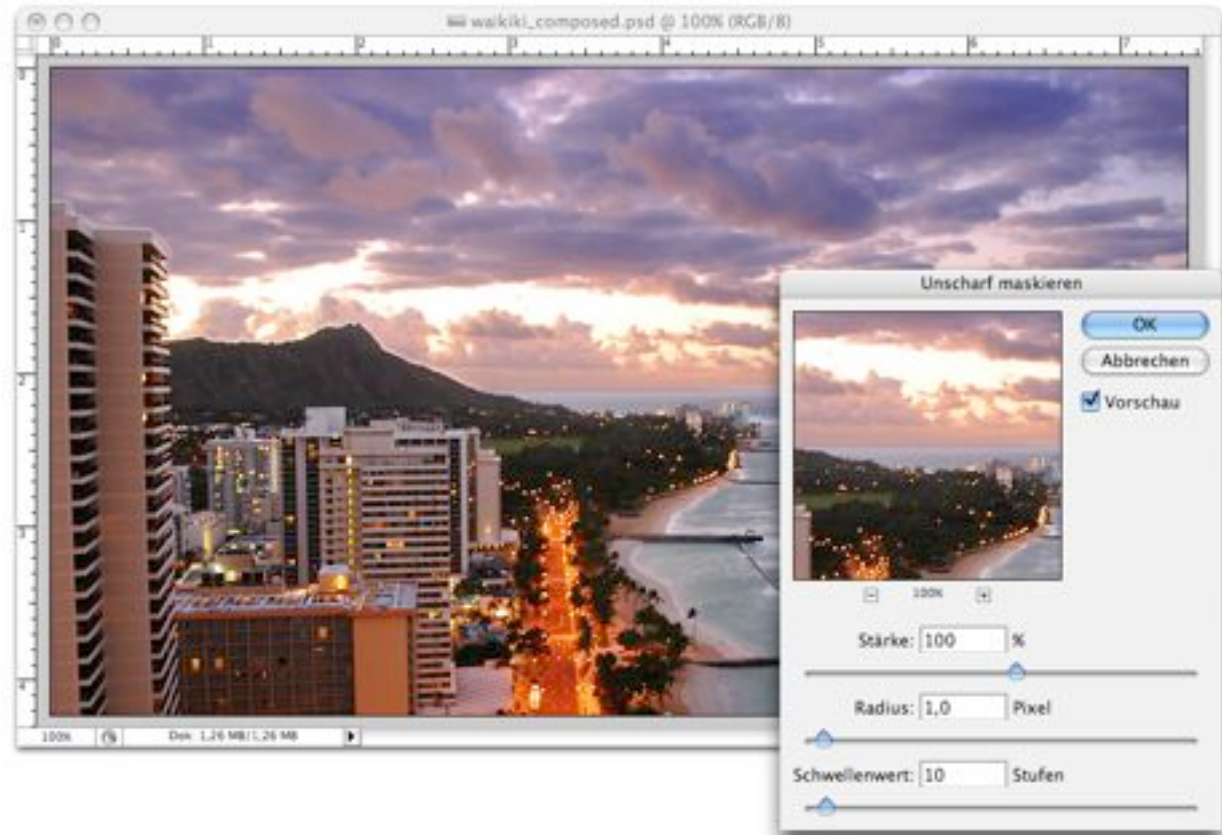


Z.B. durch Schwellwertbildung aus dem Bild selbst ableiten und dann unscharf machen

Grundlegender Arbeitsablauf der Bildbearbeitung



Schärfen



Frühere Technik mit Film hieß „Unschärf Maskieren“

Mathematische Nachbildung davon liefert immer noch sehr gute Schärfung

Ansonsten prinzipiell Hochpassfilterung!

Letzter Schritt in der Kette, Nach der Skalierung auf die Zielgröße!

Für Bildschirm eher stärker

Für Print eher schwächer

Ergebnis



Horizont gerade, Ausschnitt wählen
Himmel kräftiger, Häuser heller
Beide Korrekturen kombiniert
Größe für Bildschirmdarstellung
Schärfen
Negativrand

Schwarz-Weiß-Konversion

Die Konversion eines Farbbildes in ein ansprechendes Graustufenbild ist nicht trivial

Modusumstellung des Bildbearbeitungsprogramms genügt nicht

Einfaches Hilfsmittel bei Photoshop: Kanalmixer

Ermöglicht Bestimmung, zu welchem Anteil welcher Farbkanal einfließt

Z.B. Verstärkung des Rotkanals

Verfeinerte Hilfsmittel:

Abstimmung der Tonwertkorrektur

Spezialsoftware

Z.B. *PowerRetouche B&W Studio*

Ermöglicht u.a. Simulation des Verhaltens realer S/W-Filmtypen

Schwarzweiss im Kanalmixer



Idee: Bilde die Wirkung von Farbfiltern in der SW-Fotografie nach
Rot verstärkt Wolken
Grün macht unreine Haut glatt

Weitere Filter

Große Palette an Filtern in Photoshop, Gimp etc.

Stilisierungsfiler, Kunstfilter, Malfilter, Zeichenfilter

Nur in Ausnahmefällen zur Fotonachbearbeitung verwendet!

Verfremdung, Abstraktion z.B. für Hintergrundbilder



1. Fototechnik und digitale Bildbearbeitung

1.1 Grundlagen der Fototechnik

1.2 Digitale Fotografie

1.3 Einführung in die fotografische Bildgestaltung

1.4 Speicherung digitaler Bilddaten

1.5 Bearbeitung digitaler Bilder

1.6 Programmierung von Bildbearbeitung mit Java



Literatur:

J. Knudsen: Java 2D Graphics, O'Reilly 1999, Kap. 10

Digitale Bildbearbeitung

Bilder aus der Sicht der Informatik:

spezielle Datenstruktur (insbesondere: 2-dimensionales Array)

Bearbeitung mit verschiedenen Algorithmen möglich

- *Filter.*

Ursprünglich Begriff aus der klassischen (analogen) Fotografie

Generell: Operation, die Bild in Bild transformiert

Klassische (physikalische) Filter:

Polarisationsfilter, UV-Filter

Weich-/Scharfzeichnung

Helligkeits-, Farbfilter

Effektfilter (z.B. Sterneffekt, Kachelung)

Bildbearbeitungsprogramme bieten Vielzahl von (Software-) "Filtern"

Bsp. Adobe Photoshop, Gimp

Bildbearbeitung in Java

Frühe Java-Versionen:

- In AWT Einlesung und Anzeigen von Bildern unterstützt
- Noch keine Funktionen zur Modifikation von Bildern

Java 2D (inzwischen Bestandteil jeder Java-Standardinstallation):

- Unterstützung von 2D-Grafik, komplette Rendering-Kette
- Begrenzter Satz von Bildbearbeitungsfunktionen

Java Advanced Imaging (JAI):

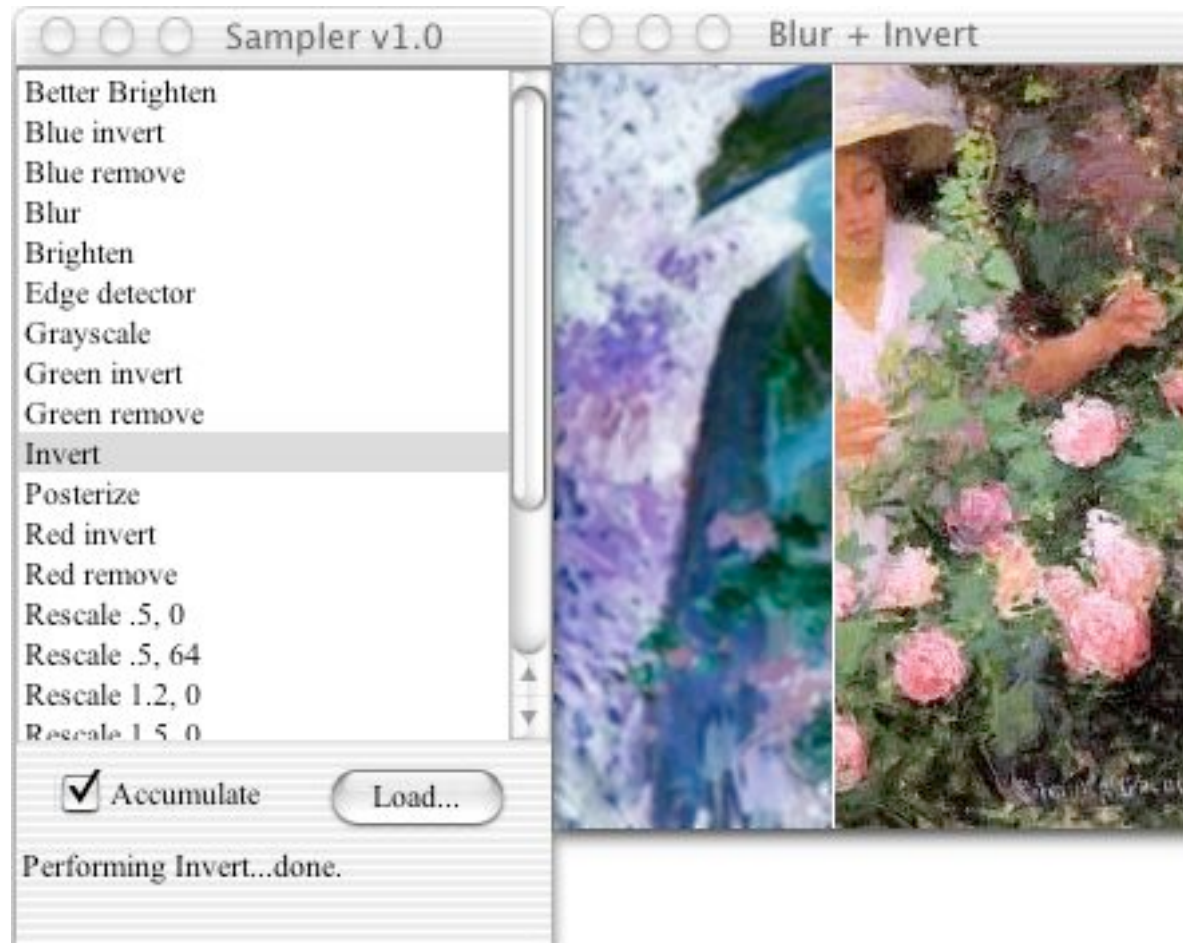
- Erste Version November 1999, aktuell: 1.1.3 (Sept. 2006)
- Ausgefeilte, hochleistungsfähige Bildbearbeitungsfunktionen
- Folgt konsequent dem Java-Prinzip "Write once, run everywhere"

Performance:

- In diesem Bereich nach wie vor das Hauptproblem der Java-Plattform
- C- und C++-Programme deutlich überlegen

Beispiel: Bildbearbeitung mit Java 2D

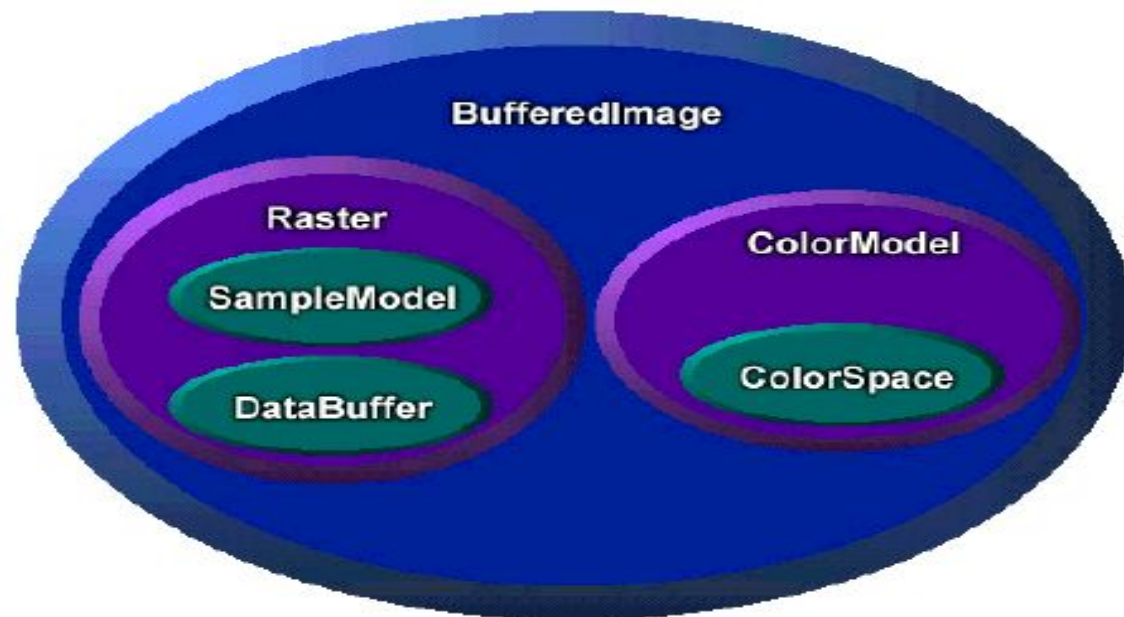
aus: Knudsen, Kapitel 10



Ethol with Roses, Edmund Greacen, 1907

Java 2D: BufferedImage

- `java.awt.image.BufferedImage`:
Repräsentation eines Bildes
Verkapselt (d.h. versteckt Details von):
 - Farbmodell
 - Abtastung
 - Datenpuffer



Einlesen von Bilddateien in Java

Einlesen von Bilddateien umfasst komplexe Algorithmen

- Decodieren des Bildformats

- Einlesen lokal aus Datei oder über eine URL

- Berücksichtigung von langsamen Festplatten- und Netzzugriffen

 - Observer-Modell: Anzeigefunktion wird wieder aufgerufen, wenn Daten nachgeladen sind

Java: Diverse Möglichkeiten zum Laden eines Bilds

- Standard-AWT-Methode (MediaTracker)

- Swing-Methode (ImageIcon)

- Spezielle Codecs (Sun-JPEG-Codec meist in Standardinstallation enthalten)

- Java Advanced Imaging

Java 2D: Bildbearbeitungsfunktionen

Bildbearbeitungsfunktionen (in Java 2D):

Schnittstelle `java.awt.image.BufferedImageOp`

```
public BufferedImage filter  
    (BufferedImage src, BufferedImage dst)
```

Bearbeitet `src`, mit genauer zu definierendem Algorithmus

Liefert bearbeitetes Bild als Resultat

`dst` ermöglicht Angabe eines Speicherbereichs für das Ergebnis

- Falls `dst = null`: neues Bild erzeugt
- `dst = src`: Operation "auf der Stelle" ausgeführt (*in place*)

Operationen werden als Objekte erzeugt

Entwurfsmuster "*Strategy*" (Gamma et al.)

Ausführung:

Entweder bei Übergabe an `drawImage()`

oder durch Aufruf der Methode `filter()` des Operations-Objekts

Java 2D: Verwendung vordefinierter Operationen

Beispiel: Konversion in Graustufen

```
public static BufferedImage convertToGrayscale  
    (BufferedImage source) {  
    BufferedImageOp op =  
        new ColorConvertOp(  
            ColorSpace.getInstance(ColorSpace.CS_GRAY), null);  
    return op.filter(source, null);  
}
```



Java 2D: Einfaches Rahmenprogramm für Operationen

```
import java.awt.*;
import java.awt.geom.*;
import java.awt.color.*;
import java.awt.image.*;
import java.io.*;
import com.sun.image.codec.jpeg.*;

public class GrayJpeg extends Frame {

    private static BufferedImage mImage;

    ... Einlesen des Bildes nach mImage ...

    public void paint(Graphics g) {
        Graphics2D g2 = (Graphics2D)g;
        BufferedImageOp op =
            new ColorConvertOp(ColorSpace.getInstance
                (ColorSpace.CS_GRAY), null);
        g2.drawImage(mImage, op, 0, 0);
    }
}
```

Vordefinierte Operationen in Java 2D

Klasse	Hilfsklassen	Effekte	"in place"? (src = dst)
<code>ConvolveOp</code>	<code>Kernel</code>	Weich- und Scharfzeichnen,	nein
<code>Affine TransformOp</code>	<code>java.awt.geom.AffineTransform</code>	Geometrische Transformationen	nein
<code>LookupOp</code>	<code>LookupTable</code> , <code>ByteLookupTable</code> , <code>ShortLookupTable</code>	Inversion, Farbtrennung, Aufhellung, Thresholding	ja
<code>RescaleOp</code>		Aufhellen, Abdunkeln	ja
<code>Color ConvertOp</code>	<code>java.awt.Color.ColorSpace</code>	Farbraum- konversion	ja

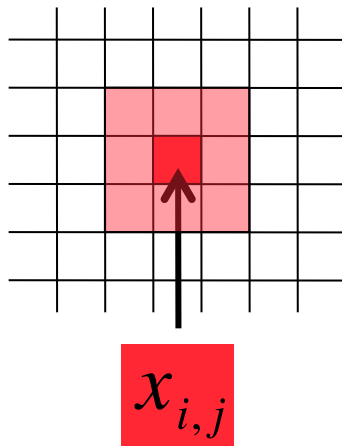
Faltung

Mathematisches Prinzip: Faltung (*spatial convolution*)

Berechnung der Farbe eines Zielpixels aus der Farbe des entsprechenden Quellpixels *und seiner Nachbarn*

Gewichtsfaktoren gegeben durch Matrix: Faltungskern (*kernel*)

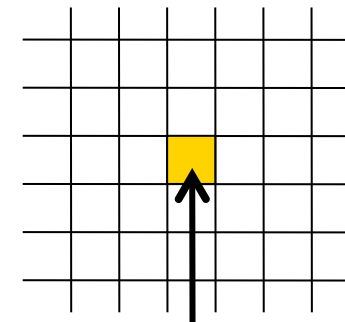
Summe der Matrixeinträge 1: Gesamthelligkeit unverändert



Quellbild

$k_{-1,-1}$	$k_{0,-1}$	$k_{+1,-1}$
$k_{-1,0}$	$k_{0,0}$	$k_{+1,0}$
$k_{-1,+1}$	$k_{0,+1}$	$k_{+1,+1}$

Matrix (*kernel*)



$$x'_{i,j} = \sum_{\substack{-1 \leq r \leq +1 \\ -1 \leq s \leq +1}} k_{r,s} \times x_{i+r,j+s}$$

Zielbild

Zusätzlich müssen die Zielwerte auf den zulässigen Wertebereich beschränkt (abgeschnitten) werden.

Mittelwertoperator: Weichzeichnen

Faltungsfiler, das Übergänge glättet ("verschmiert", *blur filter*)

- Wertverteilung im Zielbild gleichmäßiger als im Quellbild
- Gleichverteilung der Gewichte in der Matrix: bei 3x3-Matrix alle Einträge 1/9

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Java-Quellcode dazu:

```
float ninth = 1.0f/9.0f;  
float[] blurKernel = {  
    ninth, ninth, ninth,  
    ninth, ninth, ninth,  
    ninth, ninth, ninth  
};  
ConvolveOp blurOp = new ConvolveOp  
    (new Kernel(3, 3, blurKernel),  
    ConvolveOp.EDGE_NO_OP, null);
```



Java 2D: ConvolveOp

Klasse `Kernel`:

```
public Kernel (int width, int height, float[] data)
```

- Konstruiert eine neue *kernel*-Matrix mit gegebenen Dimensionen
- Das angegebene Array muss `width` x `height` viele Werte enthalten

Erzeugung des Operators

`ConvolveOp` implementiert das Interface `BufferedImageOp`

```
public ConvolveOp(Kernel kernel, int edgeHint)
```

- erzeugt einen Faltungsoperator mit gegebenem *kernel*
- Zusatzangabe zur Behandlung der Pixel an Aussenkanten
 - » `EDGE_ZERO_FILL`: Randpixel des Zielbildes werden schwarz
 - » `EDGE_NO_OP`: Randpixel des Zielbildes bleiben unverändert

Schärfen

Schärfung:

- Filter, das jedes Pixel unverändert lässt, wenn seine Umgebung den gleichen Wert wie das Pixel selbst hat
- Bei Änderungen in der Umgebung wird der Kontrast der Änderung verstärkt

Idee:

- Umgebungsgewichte negativ
- kompensiert durch Gewicht des zentralen Pixels

Beispiele:

0	-1	0
-1	5	-1
0	-1	0

-1	-1	-1
-1	9	-1
-1	-1	-1

Lookup-Tabellen

Lookup-Tabellen erlauben eine direkte Umrechnung der Werte des Quellbildes in Werte des Zielbildes

Tabellierte Funktion:

Quellwerte als Index für Tabelle benutzt, Zielwerte sind Einträge

Meist Werte zwischen 0 und 255, also 255 Tabelleneinträge

Verschiedene Varianten für Datentyp der Einträge (Byte, Short)

Beispiel: Inversion

Ähnlich zum fotografischen Negativbild

```
short[] invert = new short[256];  
for (int i = 0; i < 256; i++)  
    invert[i] = (short) (255 - i);  
LookupTable table =  
    new ShortLookupTable(0, invert);  
LookupOp invertOp =  
    new LookupOp(table, null);
```



Helligkeits-Skalierung

Globale lineare Veränderung der Helligkeitswerte

Zwei Einflussmöglichkeiten

Skalierungsfaktor (*scale factor*)

Verschiebung (*offset*)

Beispiele:

Helligkeit um 50% erhöhen

```
RescaleOp brighterOp =  
    new RescaleOp(1.5f, 0, null);
```

Helligkeit um 50% reduzieren und absolute Korrektur um 64 Schritte

```
RescaleOp dimOffsetOp =  
    new RescaleOp(0.5f, 64f, null);
```